Ling 566
Oct 24, 2013

Lexical Types

Overview

- Motivation for lexical hierarchy
- Default inheritance
- Tour of the lexeme hierarchy
- The Case Constraint
- *pos* vs. *lexeme*
- Reading Questions

Motivation

- We've streamlined our grammar rules...
  - ...by stating some constraints as general principles
  - ...and locating lots of information in the lexicon.
  - Our lexical entries currently stipulate a lot of information that is common across many entries and should be stated only once.
- Examples?
- Ideally, particular lexical entries need only give phonological form, the semantic contribution, and any constraints truly idiosyncratic to the lexical entry.

Lexemes and Words

- **Lexeme**: An abstract proto-word which gives rise to genuine words. We refer to lexemes by their ‘dictionary form’, e.g. ‘the lexeme *run*’ or ‘the lexeme *dog*’.
- **Word:** A particular pairing of form and meaning. *Running* and *ran* are different words

Lexical Types & Lexical Rules

- Lexemes capture the similarities among *run, runs, running,* and *run*.
- The lexical type hierarchy captures the similarities among *run, sleep,* and *laugh*, among those and other verbs like *devour* and *hand*, and among those and other words like *book*.

  Q: What do *devour* and *book* have in common?

  A: The SHAC
- Lexical rules capture the similarities among *runs*, *sleeps*, *devours*, *hands*,...

Default Inheritance

Q: Why do we have default inheritance?

A: Generalizations with exceptions are common:

- Most nouns in English aren't marked for CASE, but pronouns are.
- Most verbs in English only distinguish two agreement categories (*3sing* and *non-3sing*), but *be* distinguishes more.
- Most prepositions in English are transitive, but *here* and *there* are intransitive.
- Most nominal words in English are 3rd person, but some (all of them pronouns) are 1st or 2nd person.
- Most proper nouns in English are singular, but some (mountain range names, sports team names) are plural.

Default Inheritance, Technicalities

If a type says ARG-ST / < NP >,	and one of its subtypes says ARG-ST < >,	then the ARG-ST value of instances of the subtype is < >.
If a type says ARG-ST < NP >,	and one of its subtypes says ARG-ST < >,	then this subtype can have no instances, since they would have to satisfy contradictory constraints.

Default Inheritance, More Technicalities

- If a type says MOD / < S >, and one of its subtypes says MOD <[SPR < NP>] >, then the ARG-ST value of instances of the subtype is what?

$$\begin{bmatrix} \text{MOD} & \left\langle \begin{bmatrix} \text{HEAD} & / \ \textit{verb} \\ \text{SPR} & \langle \text{NP} \rangle \\ \text{COMPS} & / \ \langle\ \rangle \end{bmatrix} \right\rangle \end{bmatrix}$$

- That is, default constraints are 'pushed down'

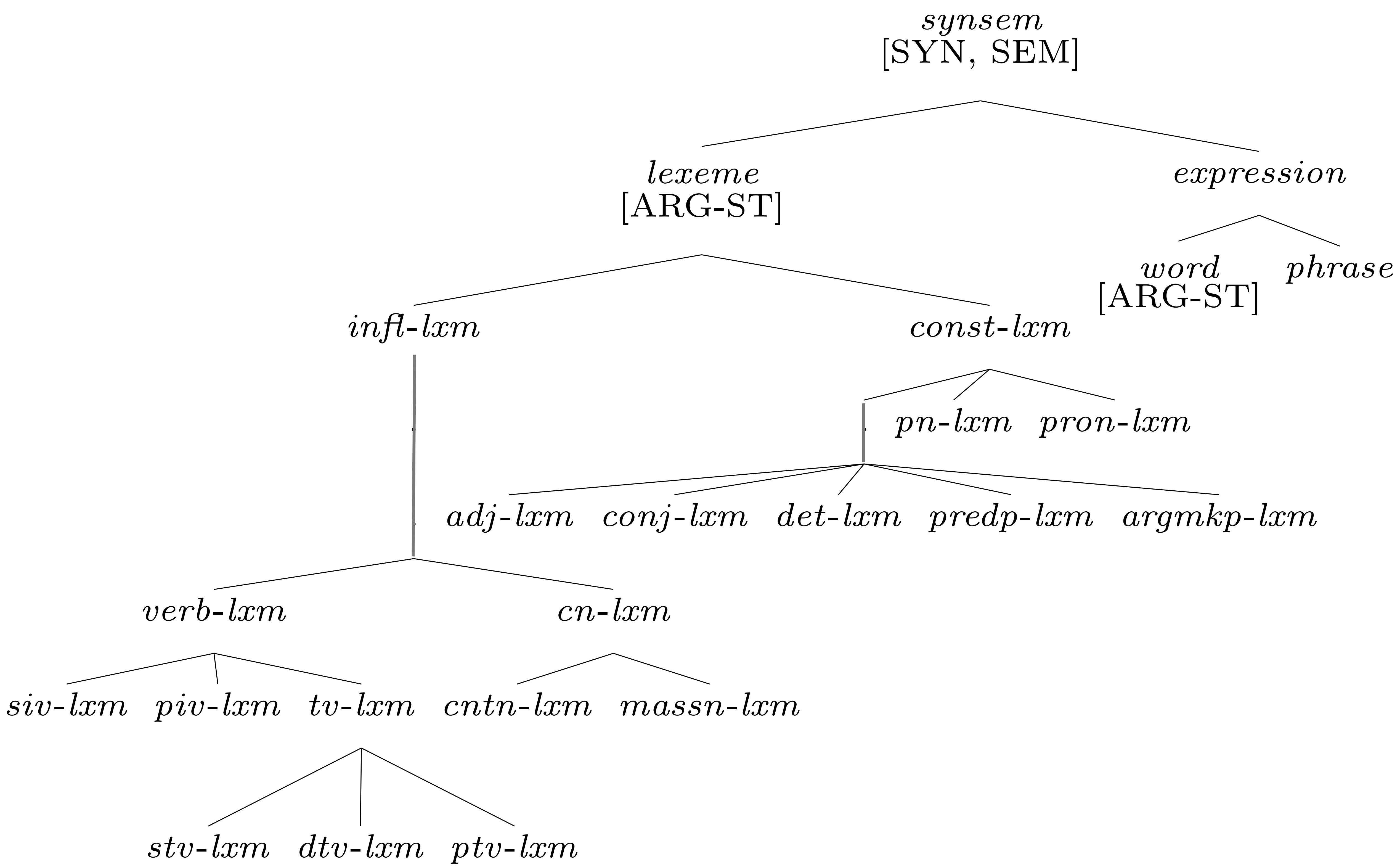
Question on Default Inheritance

Q: Can a grammar rule override a default constraint on a word?

A: No. Defaults are all ‘cached out’ in the lexicon.

- Words as used to build sentences have only inviolable constraints.

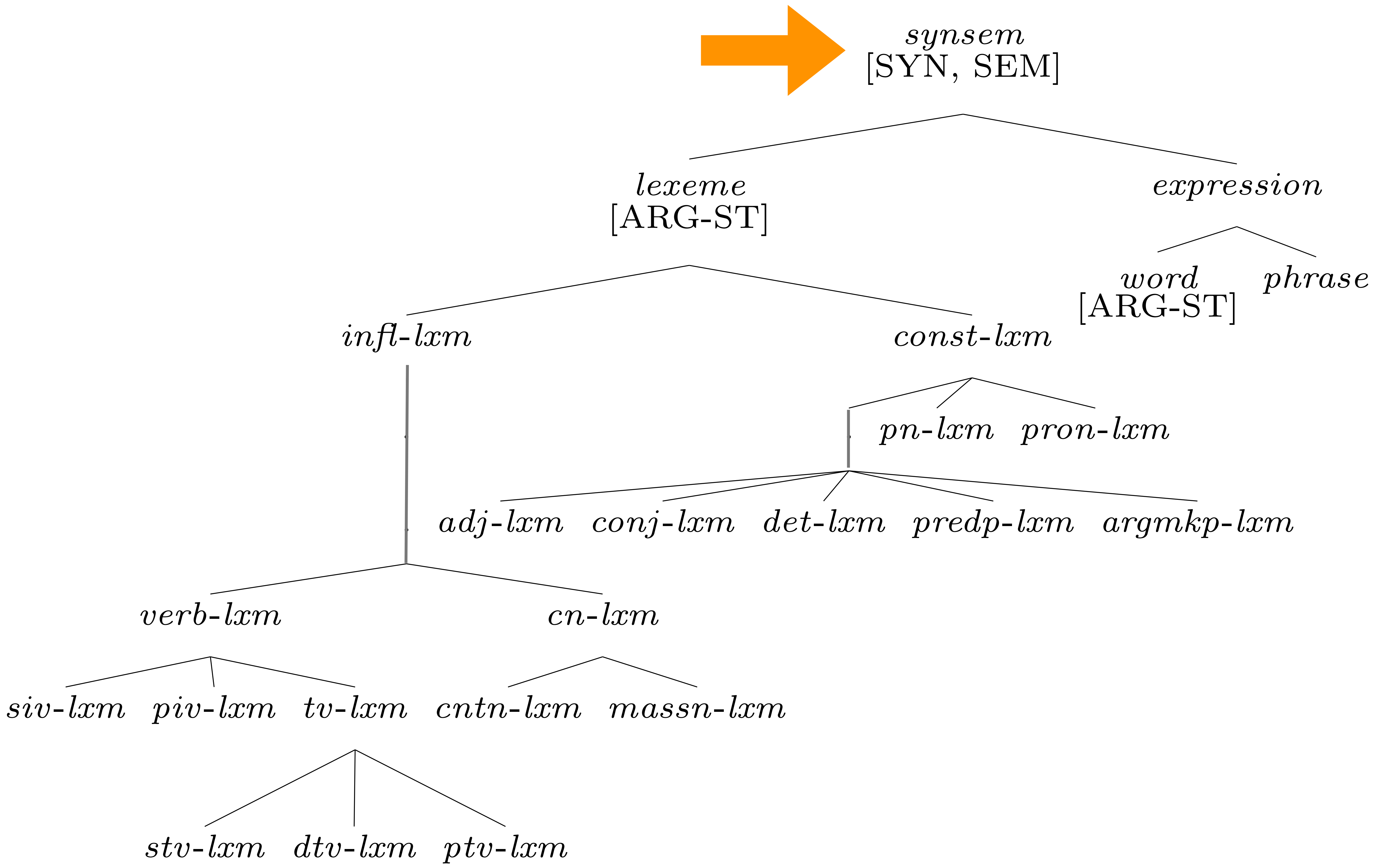
Our Lexeme Hierarchy



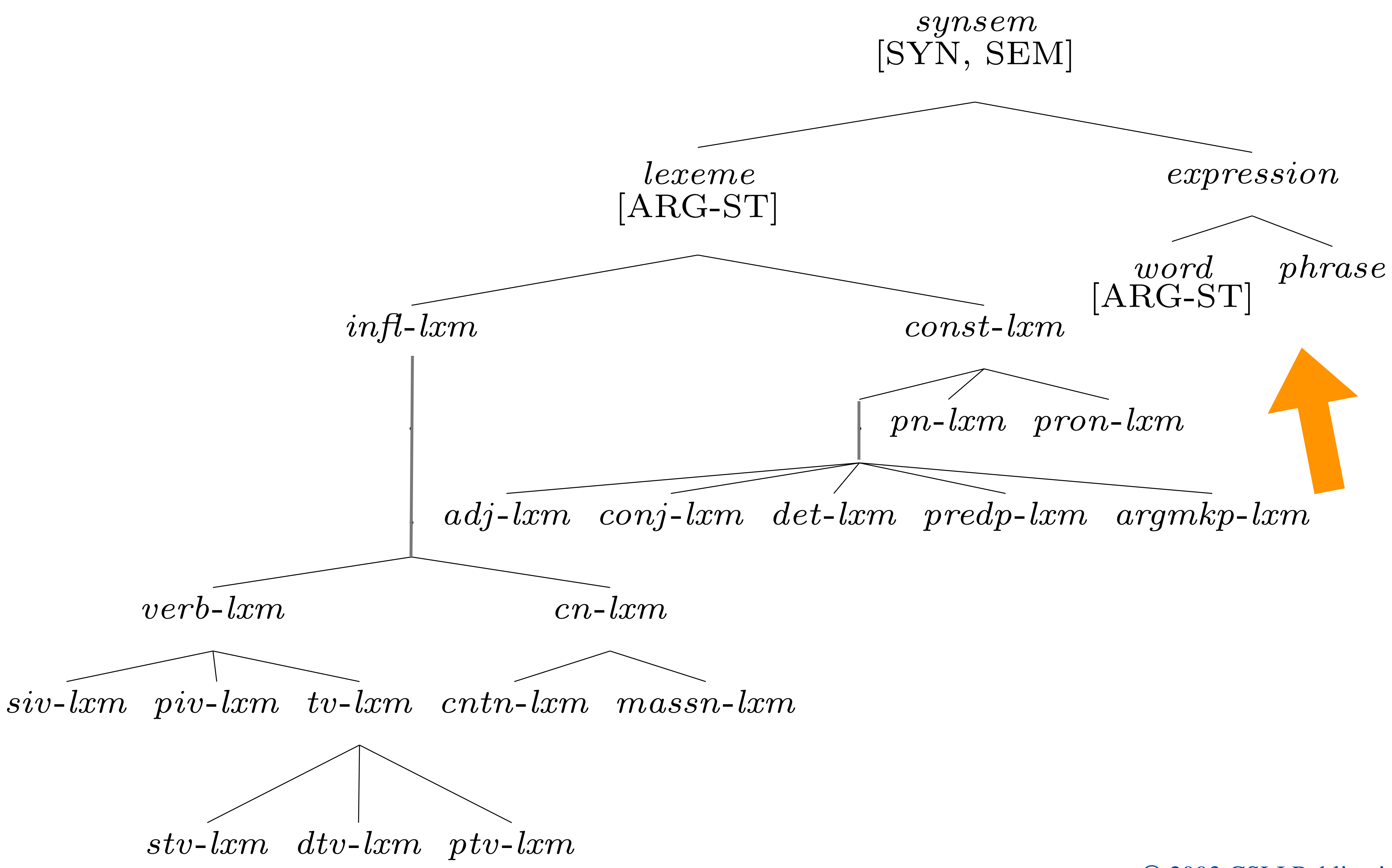
Functions of Types

- Stating what features are appropriate for what categories
- Stating generalizations
  - Constraints that apply to (almost) all instances
  - Generalizations about selection -- where instances of that type can appear

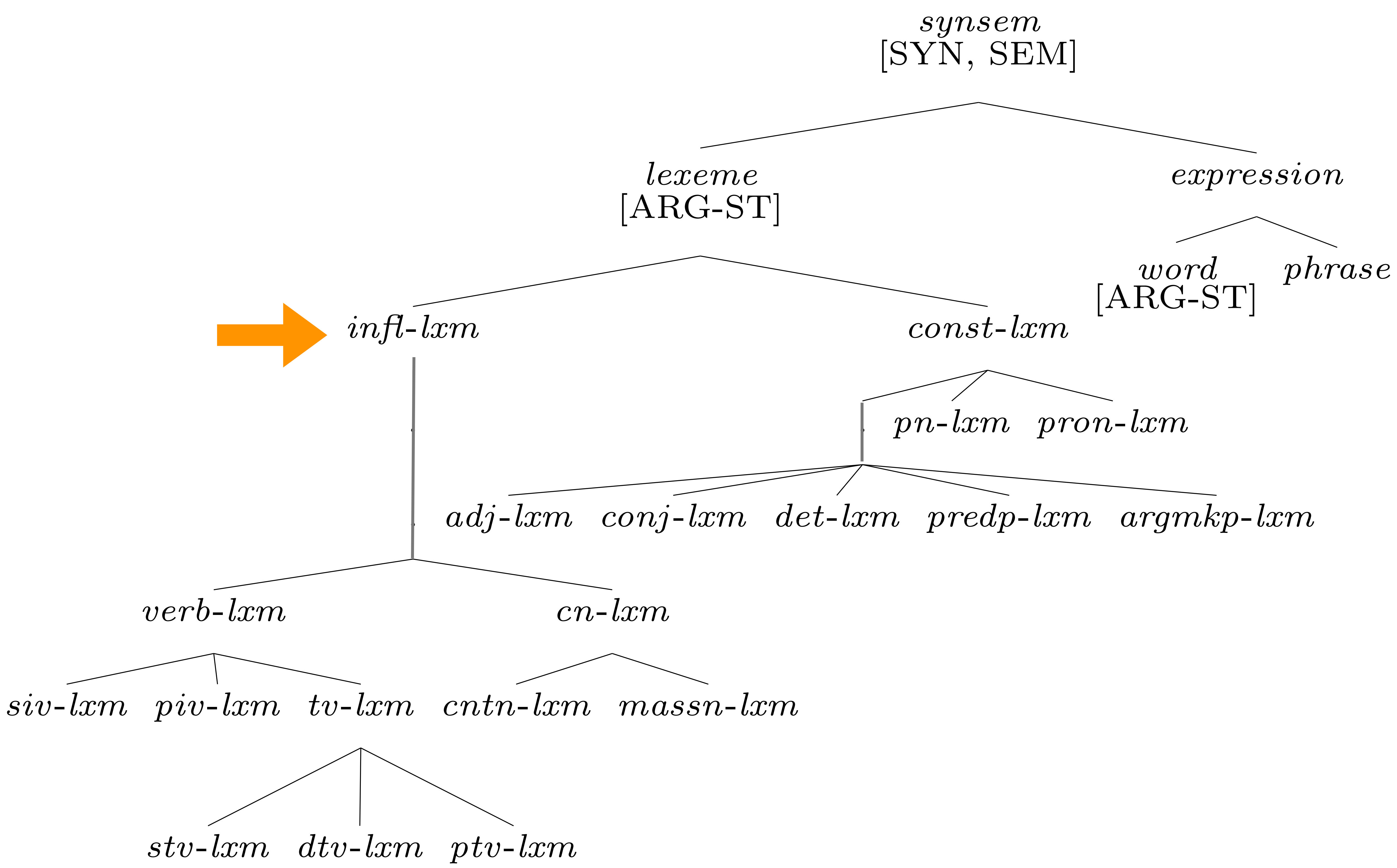
Every *synsem* has the features SYN and SEM



No ARG-ST on *phrase*



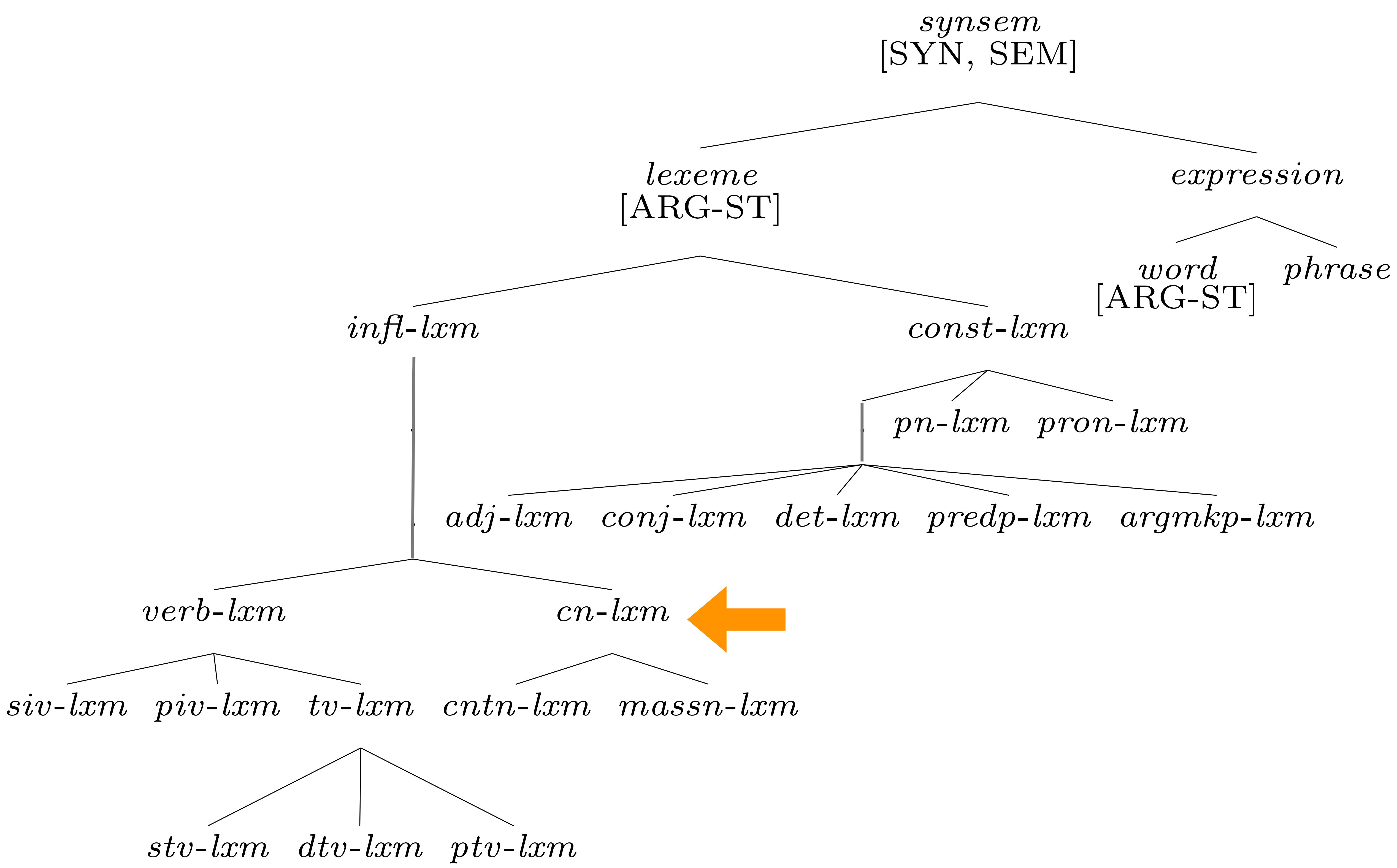
A Constraint on *infl-lxm*: the SHAC



A Constraint on *infl-lxm*: the SHAC

$$\textit{infl-lxm}: \begin{bmatrix} \text{SYN} & \begin{bmatrix} \text{VAL} & \begin{bmatrix} \text{SPR} & \left\langle \begin{bmatrix} \text{AGR} & \boxed{1} \end{bmatrix} \right\rangle \end{bmatrix} \\ \text{HEAD} & \begin{bmatrix} \text{AGR} & \boxed{1} \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

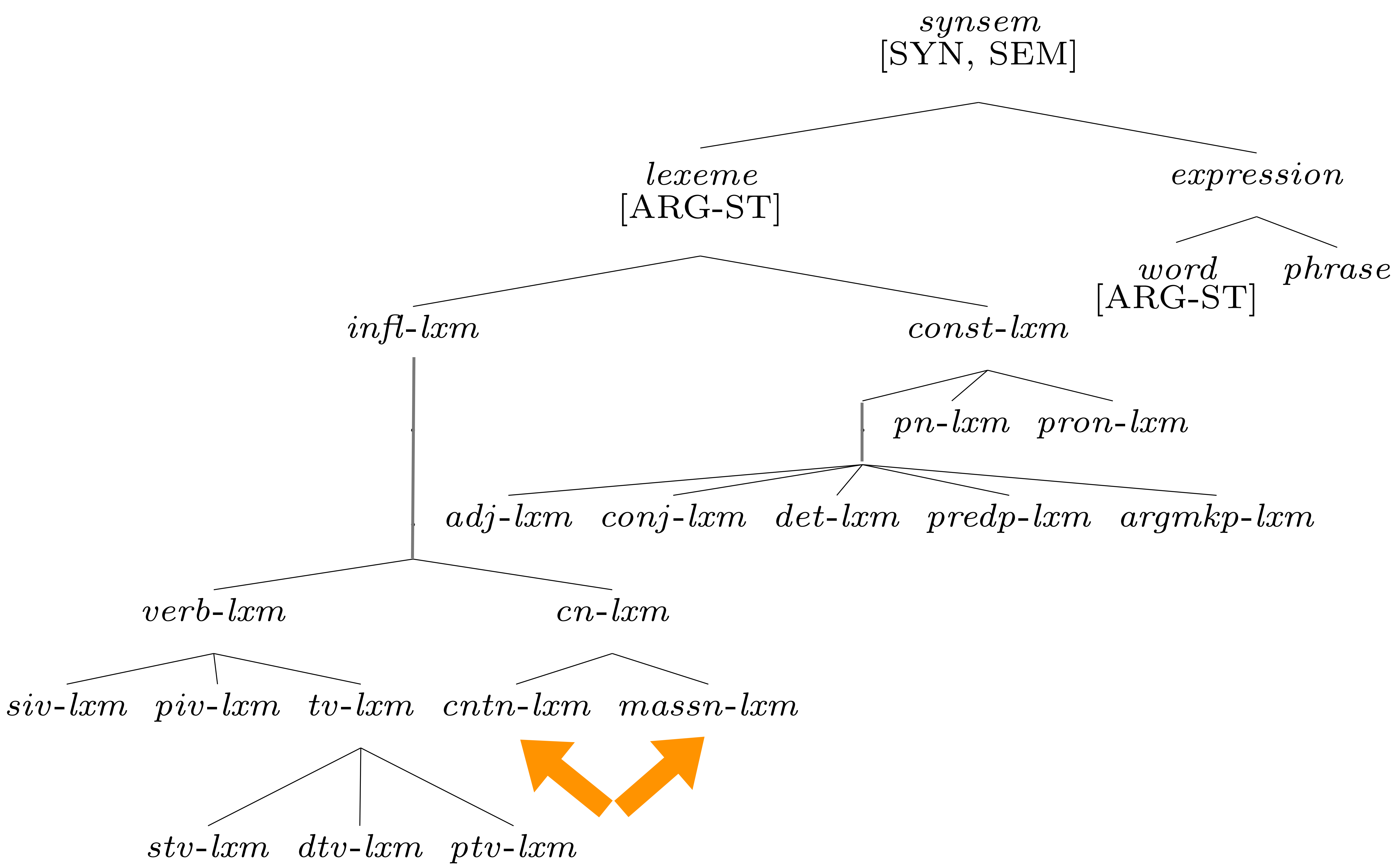
Constraints on *cn-lxm*



Constraints on *cn-lxm*

$$
cn\text{-}lxm : \begin{bmatrix} \text{SYN} & \begin{bmatrix} \text{HEAD} & \begin{bmatrix} \textit{noun} & \\ \text{AGR} & [\text{PER 3rd}] \end{bmatrix} \\ \text{VAL} & \begin{bmatrix} \text{SPR} & \left\langle \begin{bmatrix} \text{HEAD} & \text{det} \\ \text{INDEX} & i \end{bmatrix} \right\rangle \end{bmatrix} \end{bmatrix} \\ \text{SEM} & \begin{bmatrix} \text{MODE} & /\ \text{ref} \\ \text{INDEX} & i \end{bmatrix} \\ \text{ARG-ST} & \langle \text{X} \rangle \oplus /\langle\ \rangle \end{bmatrix}
$$

Formally Distinguishing Count vs. Mass Nouns

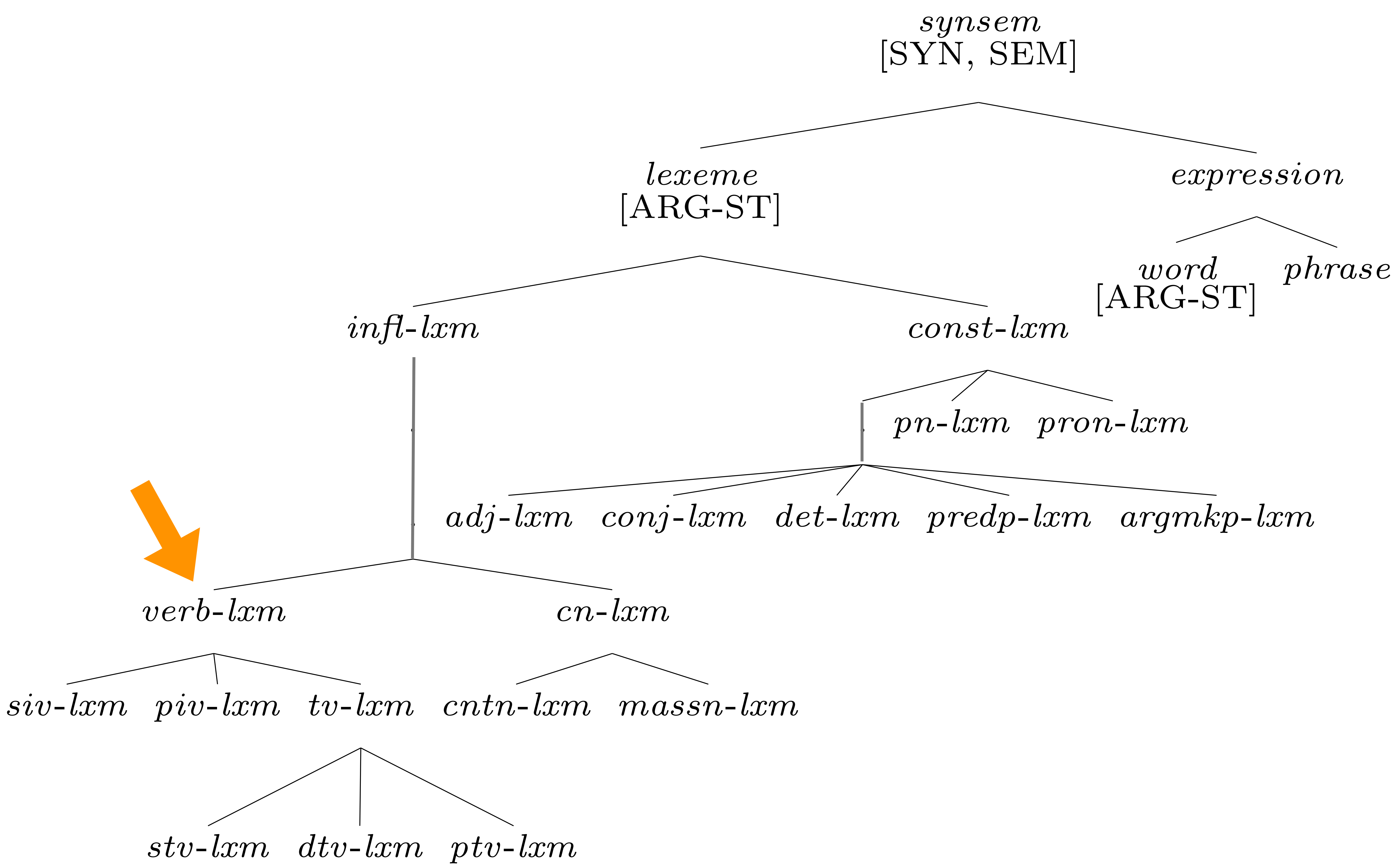


Formally Distinguishing Count vs. Mass Nouns

$$\textit{cntn-lxm} : \left[\text{SYN} \left[\text{VAL} \left[\text{SPR} \quad \langle\, [\text{COUNT} +]\,\rangle\right]\right]\right]$$

$$\textit{massn-lxm} : \left[\text{SYN} \left[\text{VAL} \left[\text{SPR} \quad \langle\, [\text{COUNT} -]\,\rangle\right]\right]\right]$$

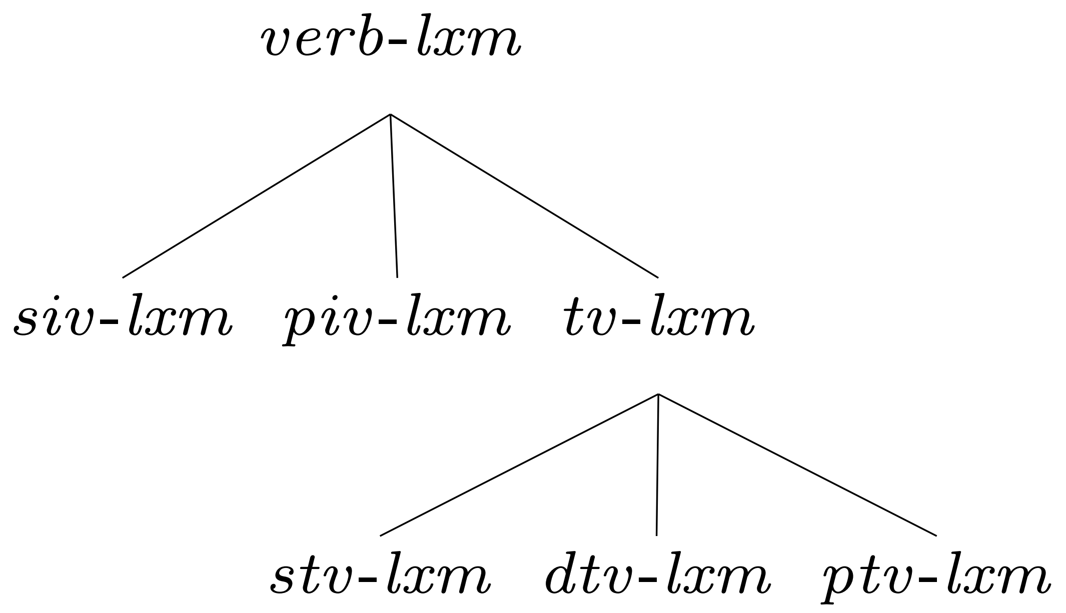
Constraints on *verb-lxm*



Constraints on *verb-lxm*

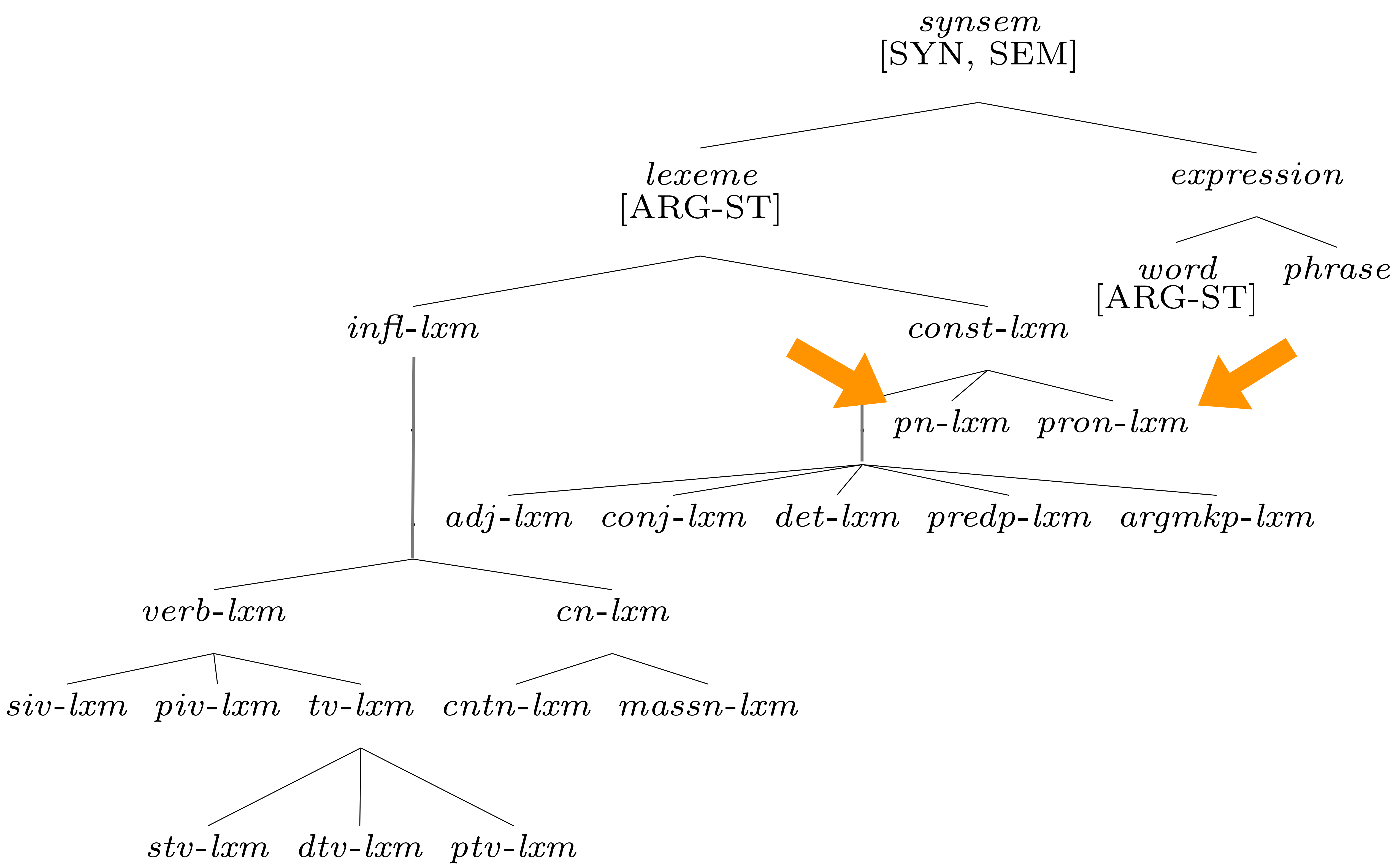
$$verb\text{-}lxm\text{:}\quad \begin{bmatrix} \text{SYN} & \begin{bmatrix} \text{HEAD} & \textit{verb} \end{bmatrix} \\ \text{SEM} & \begin{bmatrix} \text{MODE} & \text{prop} \end{bmatrix} \\ \text{ARG-ST} & /\ \langle\ \text{NP},\ \ldots\ \rangle \end{bmatrix}$$

Subtypes of *verb-lxm*



- *verb-lxm*: [ARG-ST / < NP, ... >]
  - *siv-lxm*: [ARG-ST / < NP >]
  - *piv-lxm*: [ARG-ST / < NP, PP >]
  - *tv-lxm*: [ARG-ST / < NP, NP, ... >]
    - *stv-lxm*: [ARG-ST / < NP, NP, >]
    - *dtv-lxm*: [ARG-ST / < NP, NP, NP >]
    - *ptv-lxm*: [ARG-ST / < NP, NP, PP >]

Proper Nouns and Pronouns



Proper Nouns and Pronouns

$$
\textit{pn-lxm}: \begin{bmatrix} \text{SYN} & \begin{bmatrix} \text{HEAD} & \begin{bmatrix} \textit{noun} & \\ \text{AGR} & \begin{bmatrix} \text{PER} & \text{3rd} \\ \text{NUM} & /\ \text{sg} \end{bmatrix} \end{bmatrix} \end{bmatrix} \\ \text{SEM} & \begin{bmatrix} \text{MODE} & \text{ref} \end{bmatrix} \\ \text{ARG-ST} & /\ \langle\ \rangle \end{bmatrix}
$$

$$
\textit{pron-lxm}: \begin{bmatrix} \text{SYN} & \begin{bmatrix} \text{HEAD} & \textit{noun} \end{bmatrix} \\ \text{SEM} & \begin{bmatrix} \text{MODE} & /\ \text{ref} \end{bmatrix} \\ \text{ARG-ST} & \langle\ \rangle \end{bmatrix}
$$

The Case Constraint

An outranked NP is [CASE acc].

• object of verb	✓
• second object of verb	✓
• object of argument-marking preposition	✓
• object of predicational preposition	(✓)

The Case Constraint, continued

An outranked NP is [CASE acc].

- Subjects of verbs
  - Should we add a clause to cover nominative subjects?
    - No.

    We expect them to leave. (Chapter 12)
    - Lexical rules for finite verbs will handle nominative subjects.
- Any other instances of case marking in English?
- Does it apply to case systems in other languages?

  No: The Case Constraint is an English-specific constraint.

Apparent redundancy

- Why do we need both the *pos* subhierarchy and lexeme types?
- *pos*:
    - Applies to words and phrases; models relationship between then
    - Constrains which features are appropriate (no AUX on *noun*)
- *lexeme*:
    - Generalizations about combinations of constraints

Lexical Types & Lexical Rules

- Lexemes capture the similarities among *run*, *runs*, *running*, and *run*.

- The lexical type hierarchy captures the similarities among *run*, *sleep*, and *laugh*, among those and other verbs like *devour* and *hand*, and among those and other words like *book*.

- Lexical rules capture the similarities among *runs*, *sleeps*, *devours*, *hands*,...

Overview

- Motivation for lexical hierarchy
- Default inheritance
- Tour of the lexeme hierarchy
- The Case Constraint
- *pos* vs. *lexeme*
- Reading Questions

Reading Questions

- The lexeme tree and general hierarchy just seems awfully specific to me to be generalizable. How do we capture other languages?
- Which rule or principle prevents integrating the feature structure lexeme into the parsing tree ?

Reading Questions

- Does the Case Constraint rule apply only inside individual ARG-ST, or is it applied across all elements in all ARG-STs simultaneously in the tree?
- Do we need to separate lexical entries for "give" in order to handle the dative shift?
    - I gave Susan the directions.
    - I gave the directions to Susan.

Reading Questions

- Is "lexeme" the same as "root"?
- Why do we insist on saying "family of lexical sequences" everywhere?
- Do lexical sequences always pair a form and a feature structure of type lexeme?

Reading Questions

- (p.245) "It saves us from having to assign values to features where they would do no work, for example, PER in propositions or CASE in verbs" How? It seems like PER and CASE in those cases are just underspecified.

Reading Questions

- (28) specifies the ARG-ST values for cn-lxm. I am confused as to why we have to specify an optional empty list in AGR-ST addition to the DP. I understand that their are special cases, like the "picture" example given, in which you need to account for other arguments. But couldn't a subtype of cn-lxm simply add those on it's own? What is the point of the defeasible empty list if it is going to either going to a) be overridden or b) be ignored.
  - [ARG-ST < DP > (+) / < >]

Reading Questions

- Where does the second NP in the ARG-ST for predp-lxm come from then since ARG-ST is the addition of SPR and COMPS, but I don't see where we get a COMPS?
- Does ARG-ST replace SPR & COMPS?

Reading Questions

- It seems like any principle could be reformulated as a grammar rule-and any grammar rule could be restated as a general principle (at least if a person really wanted to). So the question is what would make a linguist posit a new "principle" vs. a "rule"? Is the distinction somewhat arbitrary?